Aeronautical Systems Cente

Rapidly delivering war-winning capability

HPC Tools from ASC
11 August 2004

Jay Blair CSC
ASC MSRC
jason.blair@wpafb.af.mil

Outline

Rapidly delivering war-winning capability

- MSRC background
- ASC background
- ASC MSRC mission
- ASC MSRC resources
- Motivation
- Methodology
- Two major projects
- Visual Queue history
- Visual Queue operation
- Visual Queue demo
- Visual Queue deployment
- Visual Queue - Legacy
- Batch Engine history

Outline

Rapidly delivering war-winning capability

- BE interface
- Job monitoring / controlling
- Remote file browsing
- Security
- Supported OS
- Supported schedulers
- Current deployment
- BE demo
- BE – Legacy
- Final thoughts

MSRC Background

Rapidly delivering war-winning capability

- The High Performance Computing Modernization Program (HPCMP) was initiated in 1992 in response to congressional direction to modernize the Department of Defense (DoD) laboratories' high performance computing (HPC) capabilities

- There are four MSRCs that currently operate large HPC systems available to the entire DoD HPC community

- Each MSRC has been designated to provide a complete HPC environment, data storage, including a full range of resources including hardware, software, data storage, archiving, visualization, training, and expertise in specific computational technology areas

ASC MSRC Mission

Rapidly delivering war-winning capability

- "*Empower our customers to field and support dominant warfighting systems by delivering premier High Performance Computing (HPC) capabilities*"
- The mission statement reflects the organization's intent to actively contribute to continued U.S. battlefield superiority by providing leading-edge HPC capabilities
- The ASC HPC facility is not just a provider of computing cycles.
  - Instead, the HPC facility intends to "empower [its] customers" to provide "High Performance Computing *capabilities*" that include a vast array of services in a collaborative environment that includes Government, industry, and academia

ASC MSRC Mission

Rapidly delivering war-winning capability

- The ASC HPC personnel developed a set of strategic goals that map directly to the ASC goals of product, people, technology, business, and community
  - Surpass customer needs
  - Attract, grow, and retain a quality High Performance Computing workforce
  - Advance High Performance Computing technological superiority and leading-edge computing environment
  - Improve business practices
  - Expand the opportunities for Outreach
- *We will continue to focus on the customer and do everything possible to meet and exceed their needs*

ASC MSRC Resources

Rapidly delivering war-winning capability

SGI Origin 3900

- 4 Nodes / 2048 Processors
- 2048 Gigabytes Memory
- 25.5 Terabyte Workspace
- LSF 6.0

HP SC40/45

- 225 Nodes / 900 Processors
- 900 Gigabytes Memory
- 8 Terabyte Workspace
- LSF 6.0

ASC MSRC Resources

Rapidly delivering war-winning capability

IBM P3

- 132 Nodes / 528 Processors
- 528 Gigabytes Memory
- 2.4 Terabytes Workspace
- Load Lever / Maui

Motivation

Rapidly delivering war-winning capability

- 2003 HPCMP User Group Conference
  - Minimal acknowledgements in presentations
  - Lost our “edge”
  - Something needed to change

- Epiphany - “It’s about the users, stupid”
  - *We will continue to focus on the customer and do everything possible to meet and exceed their needs.* – ASC Mission Webpage

Motivation

Rapidly delivering war-winning capability

- Steve Wourms ASC MSRC Director
  - "We're about happy computing"

- Put user in the center of our business
- Make HPC easy / fun / desirable / sexy
- ASC MSRC needs to be indispensable

- We don't "do" Windows but, it's "ok" if you do
  - Solutions needed to be truly cross platform including MS Windows

Methodology

- Rigorous problem definition combined with fresh and innovative approaches
- Software engineer on staff
- Use a team approach
  - It's really "who" not "what" you know
  - Don't reinvent the wheel
  - "Stand on the shoulders of giants"
- The road to success is truly paved with "We looked at that" and "It's been tried before"

Two Major Projects

Rapidly delivering war-winning capability

- Visual Queue - a web-based application that allows users and administrators of HPC systems to visualize the life cycle of compute jobs from the time the jobs are submitted by a user to the time they finish running and leave the system.
- Batch Engine (BE) – a user interface to simplify the creation and submission of HPC jobs through the use of user customizable templates.

Visual Queue

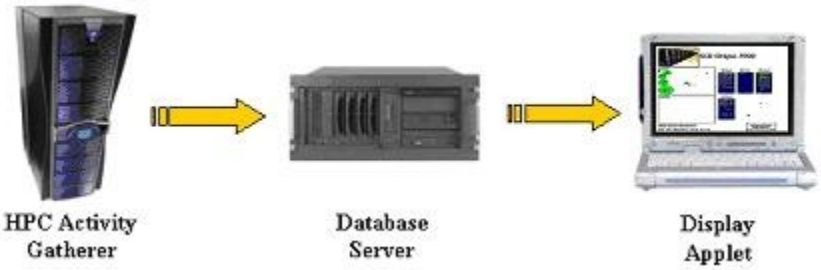
- Java application served off of the ASC MSRC Kerberized web server
- Allows users, operations, helpdesk to monitor the HPC job environment both historically and in real time
- Interfaces with batch scheduler (LSF)
- Collaborative effort of a multi-disciplinary team headed by Randy Pargman
- Randy coded it all over a 1 year timeframe

Visual Queue History

- 2002 Steve Wourms, the new ASC MSRC director
  - wanted to “see” how we were doing
  - Utilization numbers and interrupts were tough to digest
- First concepts were to “mockup” something like a 3D bar chart in Excel
- Several rapid white board engineering sessions resulted in a client side Python application that uses shapes, sizes and colors
- Once the final features were decided, Randy rewrote the application in Java

Visual Queue Operation

- The Visual Queue software consists of three distinct parts that work together to bring the visual display to the end user:
  - HPC Activity Gatherers run on each HPC system you wish to monitor, reading through the scheduler logs once per minute and sending the information to the Visual Queue Database Server.
  - The Database Server uses a commercially available or open source relational database management system (DBMS) such as Oracle or MySQL to store information about jobs, machines, queues and scheduling events. The DBMS works in conjunction with a web server (such as Apache) to allow remote users to query the database and receive responses in XML.
  - Display Applets run on each end-user's (viewer's) computer and handle all the animation and display processing. The Display Applet uses Java technology to embed the display on a web page, so there is nothing for the end user to download and install, provided they have an up-to-date web browser with Java enabled.



Visual Queue

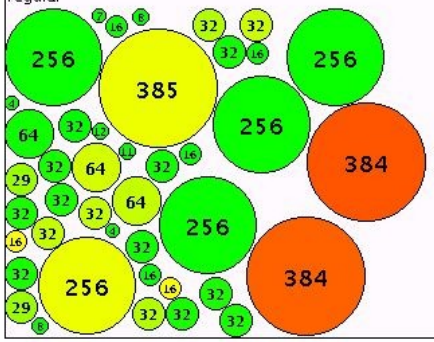
Rapidly delivering war-winning capability

Demo

SGI Origin 3900

challenge

regular



high_priority | debug | squeue | background | urgent

Server Time: Wednesday 4 Aug 04 10:28:08

hpc11-0
0 % busy

hpc11-1
99 % busy

hpc11-2
99 % busy

hpc11-3
90 % busy

hpc11-4
92 % busy

Total System Utilization
96% Busy
1942 of 2032 processors busy

0 hours — Queue Wait Time (in hours) — 48+ hours

0% done — Run Time vs. Wallclock Limit — 100% done

Visual Queue Deployment

Rapidly delivering war-winning capability

- All current ASC MSRC users have access via the ASC Kerberized website
- Operations staff monitors it real time
- Helpdesk / User Services has it displayed on LCDs mounted on wall above common printer
- Delivered code and installation support to FNMOC

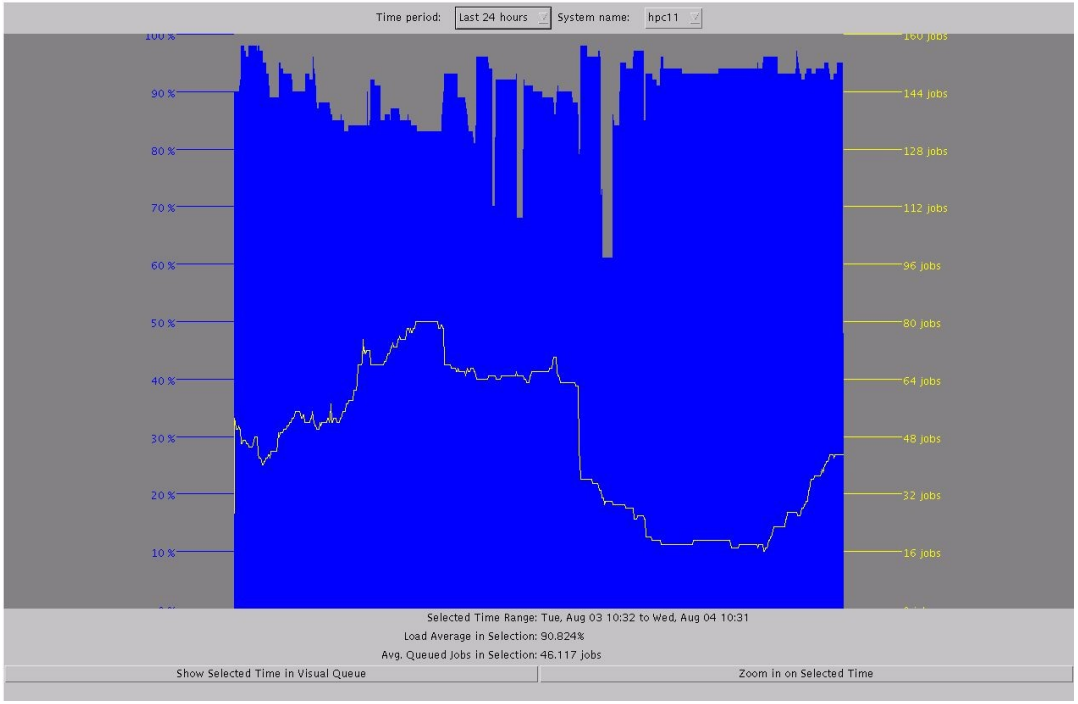
Visual Queue Legacy

- Visual Queue has become a mainstay in all tours of the ASC MSRC
  - Most popular stop / highest number of questions
- It has generated new discussions in
  - Operations / Monitoring
  - Sys Admins
  - Scheduling and Queuing
  - Helpdesk
- There have been numerous legacy projects

Visual Queue Legacy

Rapidly delivering war-winning capability

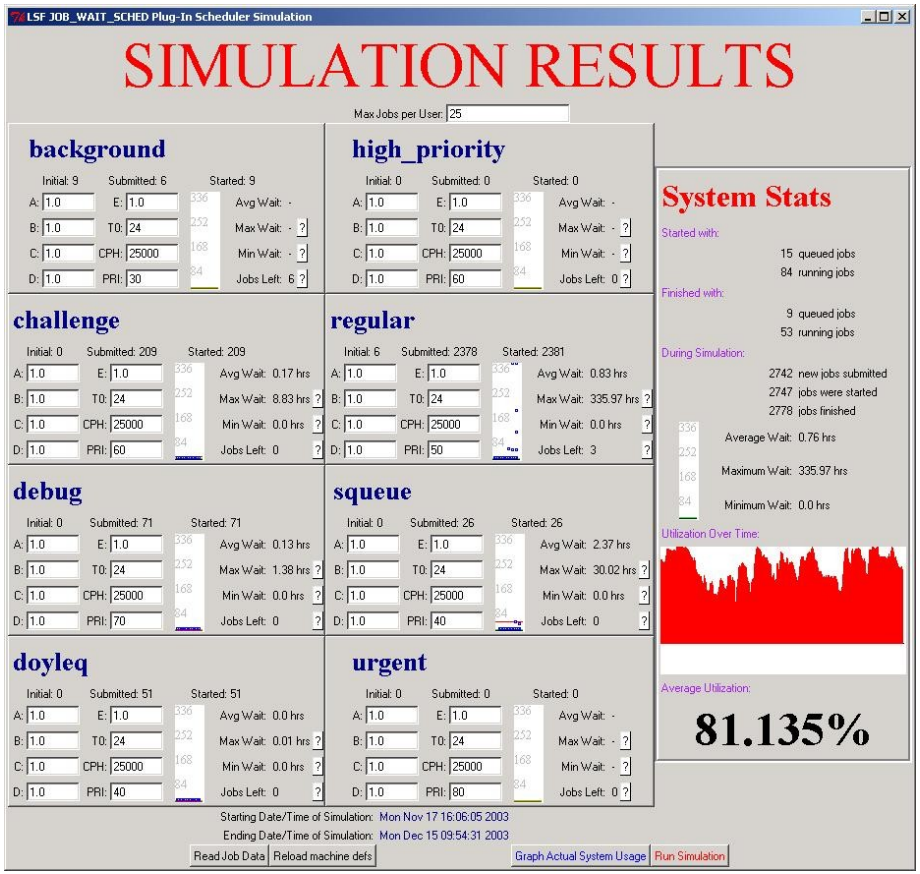
- Load Average Display
  - A Java Applet that can display the utilization and number of jobs queued over time as a graph.



Visual Queue Legacy

Rapidly delivering war-winning capability

- LSFsim
  - An application that allows the Scheduling & Queuing Team to run “what-if” scenarios with scheduler policies.
  - Uses historical data from Visual Queue



Visual Queue Legacy

Rapidly delivering war-winning capability

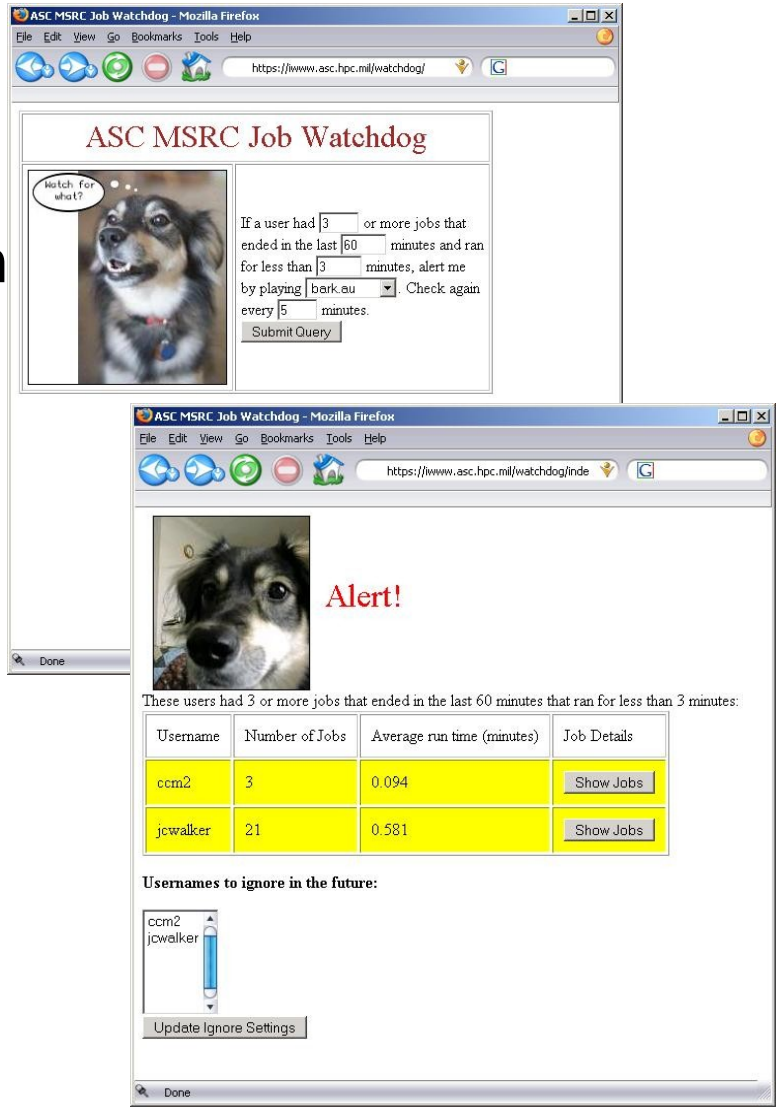
- ASC Plug-in Scheduler
  - A Platform written plug-in scheduler that implements an age based scheduling mechanism on ASC's design

$$Age = A*(initial\_queue\_priority) + \min\{1, \text{int}(\frac{T}{T_0})\}*\{B*requested\_processors + C*T*\max\{1(+\frac{1}{T_r}), 5\} + D*requested\_memory\}$$

Visual Queue Legacy

Rapidly delivering war-winning capability

- Job Watchdog
  - Web-based job monitoring program that alerts the help desk when a user has many jobs that die or run for too short a time.



Visual Queue Legacy

Rapidly delivering war-winning capability

- RSS Notification
  - RSS (Really Simple Syndication) alerts the user of “lightly” loaded machines
  - Subscription based, not SPAM!

Batch Engine (BE)

- BE is an extensible, easy-to-use GUI to simplify HPCMPO user's daily job throughput
- Built with HPCMPO distributed client kits
- No new servers or ports
- Kerberos ticket life is not limitation
- All commands are run as the Kerberos-authenticated user
- Built on Python 2.3 and Tk
- Built-in LSF and Loadleveler support
- Based on user-modifiable templates

BE History

- Surveyed existing HPCMP tools
  - Grid Engine Script Tool (GEST) from ARL
    - http://www.arl.hpc.mil/docs/gest/index.html
  - ERDC batch script generator
    - http://www.wes.hpc.mil/cgi-bin/batchwiz.cgi
  - NAVO script generator
    - http://www.navo.hpc.mil/us/FAQ/qprep_script_gen.html
- All were lacking in a least one area
  - GEST users have to be logged into ARL
  - ERDC, NAVO tools just made scripts that user had to copy and then use once logged in (no submittal)
  - None could send or receive files or check on job
  - We needed desktop support that included MS Windows

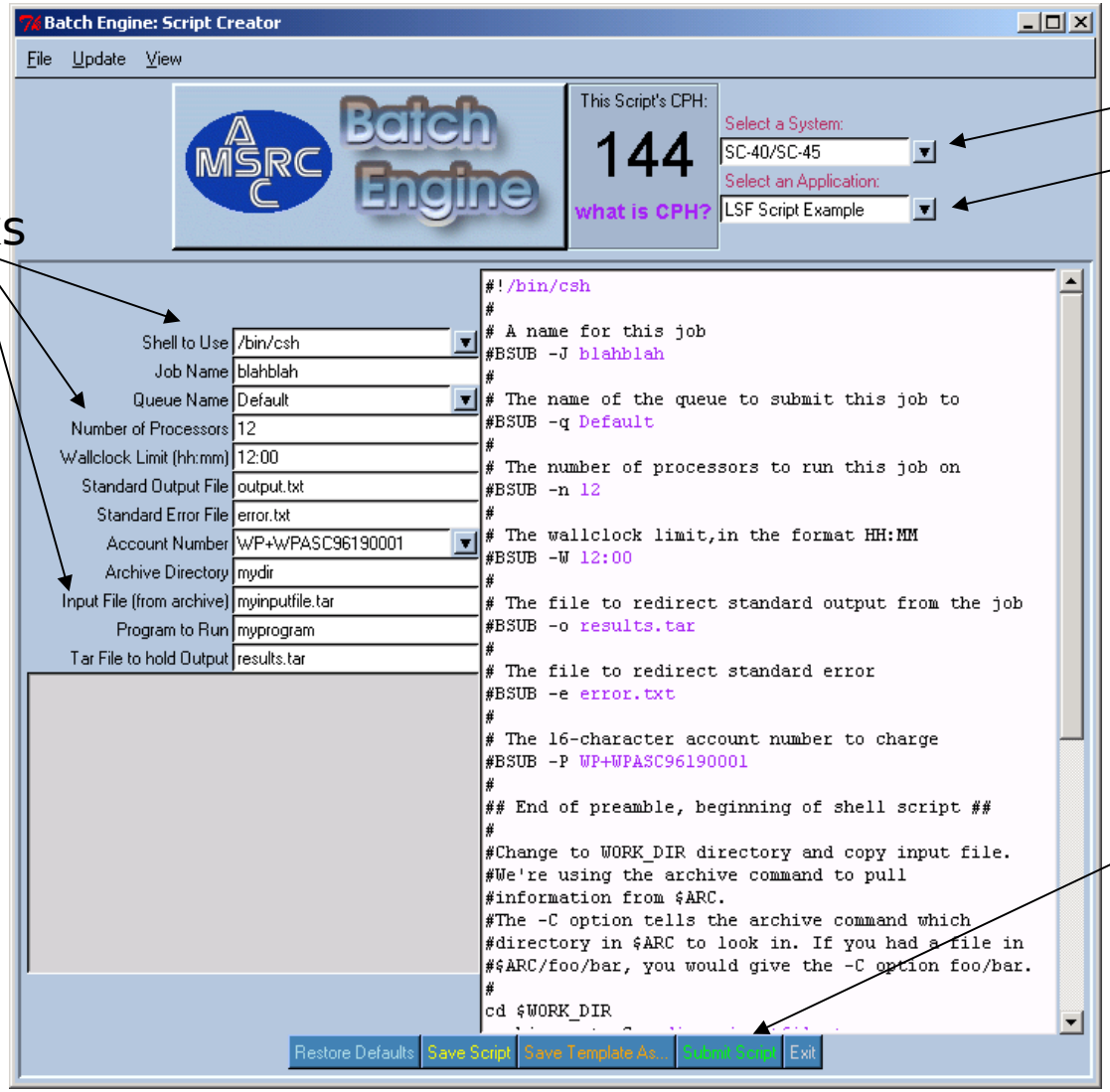
BE - Interface

Rapidly delivering war-winning capability

Step 1:
Choose system
and application

Step 2:
Fill in the blanks

Step 3:
Submit Script

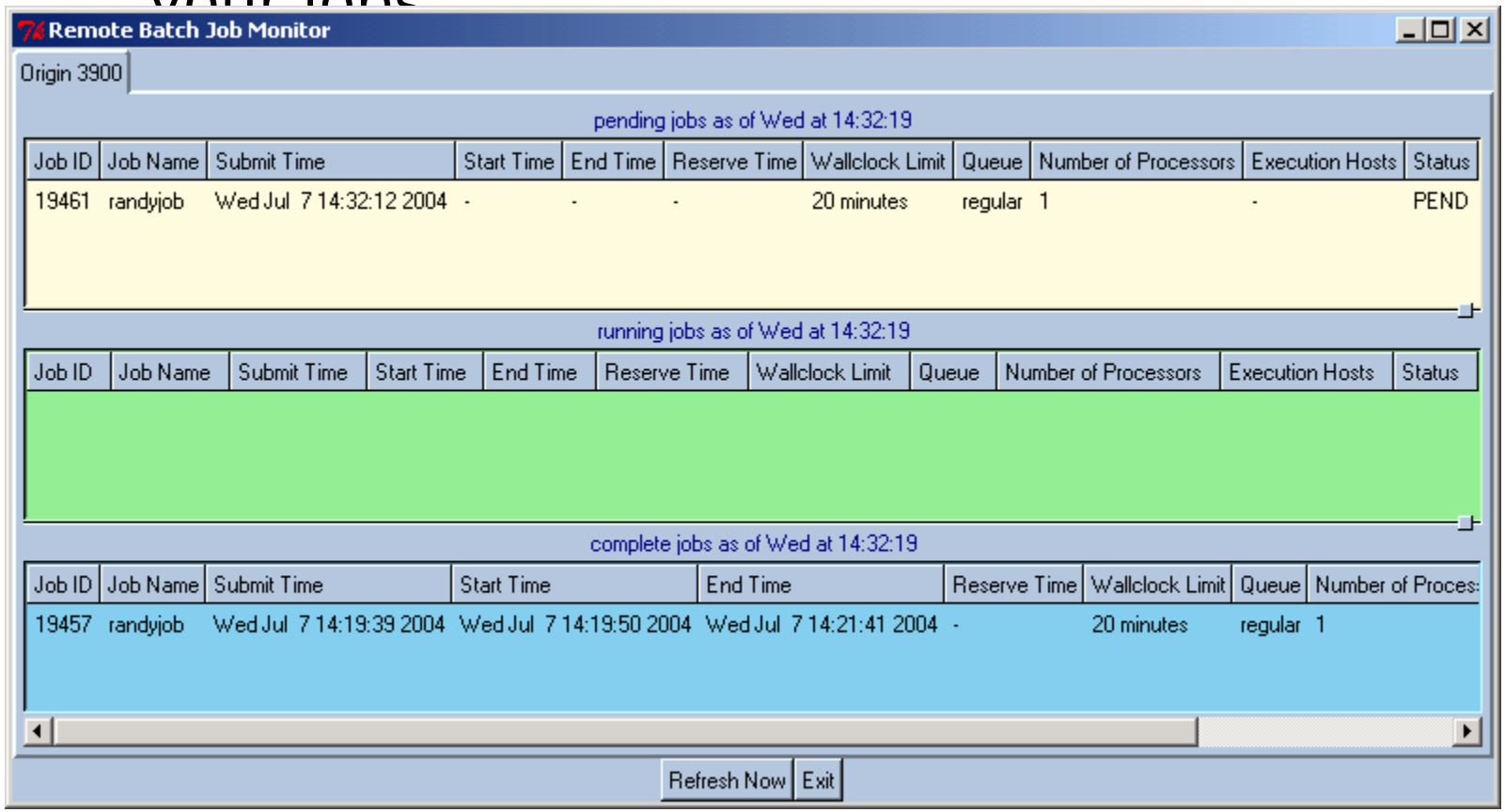


Job Monitoring/Controlling

U.S. AIR FORCE

Rapidly delivering war-winning capability

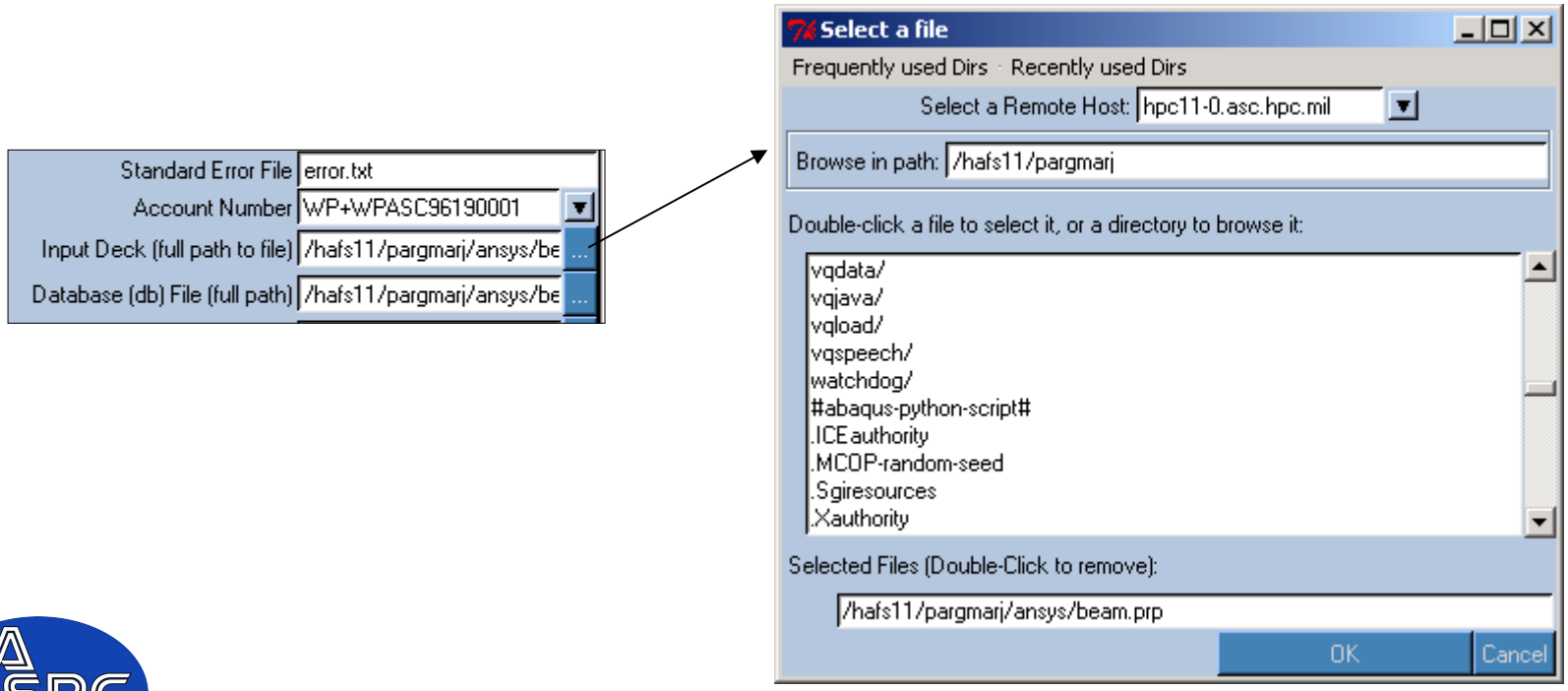
- Once a job has been submitted, you can monitor and control (suspend/resume/cancel) your jobs



Remote File Browsing

Rapidly delivering war-winning capability

- If a script requires a full path to some file, Batch Engine makes it easy to select the file by point-and-click browsing of the file system on a remote HPC machine.

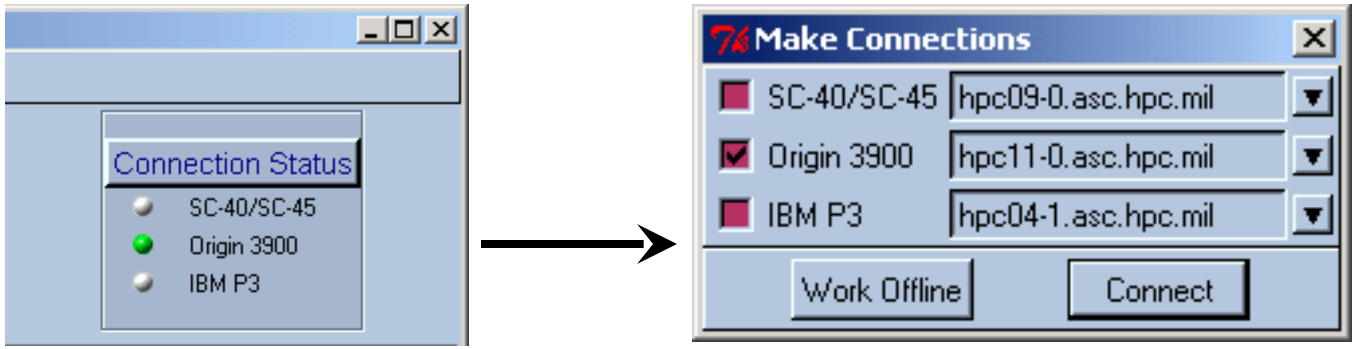


Connections to HPC Machines

U.S. AIR FORCE

Rapidly delivering war-winning capability

- Batch Engine uses existing, Kerberized SSH clients and servers to communicate between users and HPC machines
- SSH and kinit clients were obtained from kirby.hpcmo.hpc.mil
- Connections are set up, maintained and closed through a simple graphical user interface



Kerberos Tickets

- Uses Kerberos tickets to establish initial SSH connection
- A restricted command handler is run as the user on the remote HPC machine and communicates with the client over the SSH session
- Once the initial SSH connection is established, the ticket can be destroyed
- Thus the life of the Kerberos ticket is not an issue for users

The Command Handler

U.S. AIR FORCE
Rapidly delivering war-winning capability

- The Command Handler simply reads XML commands from standard input, takes some action (for valid commands) and returns an XML response on standard output
  - Stdin/stdout goes through SSH
- It is started by the user and runs completely AS THE USER (no setuid!)
- It DOES NOT listen on any port or accept connections or create any sockets

Cross Platform Client

- GUI is currently in Beta on the following platforms
  - Windows 2000, XP
  - Linux
  - IRIX
  - Tru64
  - AIX
- Requires Python 2.3 and TK
- Windows clients have self-extracting installer that includes dependencies

Different Schedulers? No problem!

- When the command handler starts, it determines what machine it is running on, and therefore, what scheduler it is supposed to use
- The Batch Engine code contains mappings that tell the command handler how to run commands on different schedulers
- For example, to submit a job on LSF, you use ‘bsub < filename’ and on LoadLeveler, you use ‘llsubmit filename’. These differences become transparent to the user in Batch Engine!

Different Schedulers Continued

- The scheduler commands currently implemented include:
  - LSF: bhist, bresume, bhosts, bkill, bstat, bsub, bjobs, bpeek, bstop
  - LoadLeveler: llsubmit, llcancel, llq, llstatus, llhold
- Framework is extensible in that any of the batch scheduler commands can be added

Current Deployment

Rapidly delivering war-winning capability

Demo

- Product is undergoing final beta testing
- Several local (WPAFB) users as well as users at the Air Force Academy in Colorado Springs
- HPCMP User Interface Tool team is evaluating the usefulness of tools of this nature
- Current 11 similar efforts within the HPCMP
- Our plan is to fold the two other efforts at ASC into ours and then distribute widely at ASC
- Leverage SIP and FMS PET to distribute to wider HPCMP audience

BE Future Work

- Signal Imaging and Processing PET requested that BE work with UAS access
  - Beta version has the latest and greatest SSH kit
- Force and Modeling Simulation PET has requested range capability to generate DOE style of jobs (thousands)
- Users have requested ability for both sending and receiving of data

Final Thoughts

Rapidly delivering war-winning capability

- ASC is committed to the user
- ASC is focusing on being “King of the Hill”
- We are adding capability to ease the use of HPC resources
- Fresh innovative approach to solving “old” problems
- Perseverance through dogged determination
  - Let people know that you ARE going to be performing a task, invite them to help